

## Decomposição espectral parcial de uma matriz em uma variedade linear

**Licio H. Bezerra**

UFSC - Departamento de Matemática  
Campus Trindade  
88040-900, Florianópolis, SC  
E-mail: licio.bezerra@ufsc.br.

**Resumo:** *Este trabalho introduz um método de ponto fixo que converge quadraticamente para os zeros da função  $F(X) = AX - XY^H AX$ , com  $X, Y \in \mathbb{C}^{n \times p}$ , na variedade dada por  $Y^H X = I$  ( $Y$  fixo). Uma das vantagens desse método em relação ao método de Newton é que todas as operações envolvidas no processo são puramente algébricas. Não há resolução de sistemas lineares como no caso do método de Newton, em que se resolve, a cada passo  $k$ , o sistema  $F'(X_k)H = F(X_k)$ .*

**Palavras-chave:** *Métodos de Ponto Fixo, Métodos de Newton, Autovetores*

A busca por novos métodos de computação parcial do espectro de uma matriz pela comunidade científica é incentivada pelas necessidades práticas de várias áreas científicas. Por exemplo, em Análise de Estabilidade Local de um sistema dinâmico, procuram-se autovalores com parte real positiva ou que possuem argumento próximo a  $\pm\pi/2$ . Em Processamento de Sinais, é usual o cálculo de autovalores dito dominantes, isto é, autovalores cujo valor absoluto são relativamente maiores quando comparados aos restantes.

O nosso objetivo aqui é introduzir um novo método que computa um subespaço invariante que está contido em uma dada variedade linear. Esse método foi desenvolvido a partir de outro método, cuja análise foi realizada em [2]. Iniciamos o nosso trabalho apresentando um breve resumo da análise desse método e reescrevemo-lo usando a notação de produto de Kronecker. O resultado é uma iteração de Newton-Raphson da forma  $V_k = V_{k-1} - f'(V_{k-1})^{-1}f(V_{k-1})$ , em que  $f'(V_{k-1})$  é uma matriz. Introduzimos, então, o nosso método, que é uma iteração do tipo Newton-Moser ([3]), que aproxima a inversa da matriz concomitantemente à evolução do método de Newton. Além disso, sabe-se que sua convergência local também é quadrática ([5]) e mostra-se que esse novo método também preserva a variedade linear, no sentido que todos os vetores resultantes das iterações pertencem à variedade, se o vetor inicial for escolhido dentro dela.

Seja  $A \in \mathbb{C}^{n \times n}$  uma matriz inversível. Vamos supor que o espectro de  $A$  é simples, isto é, que todos os autovalores de  $A$  são distintos dois a dois. Seja  $Y \in \mathbb{C}^{n \times p}$ ,  $p \leq n$ , uma matriz de posto completo. Seja  $\mathcal{X}_Y = \{X \in \mathbb{C}^{n \times p} \mid Y^H X = I\}$ . Observe que todas as matrizes de  $\mathcal{X}_Y$  também são de posto completo. Vamos considerar agora a função  $F$ , definida em  $\mathcal{X}_Y$  por

$$F(X) = AX - XY^H AX \quad (1)$$

A derivada de  $F$  em  $X$  é dada por

$$F'(X)[V] = (I - XY^H)AV - VY^H AX. \quad (2)$$

**Observação 1.** *Vamos supor que o espectro de  $A$  é igual a  $\{\lambda_i \mid i = 1 : n\}$ , todos eles distintos e não nulos. Uma solução de  $F(X) = 0$  é uma matriz  $X^*$ , de posto completo, cujas colunas geram o espaço de autovetores de  $A$  associados aos  $p$  autovalores da matriz  $Y^H AX^*$  que, sem perda de generalidade, podemos supor que são  $\lambda_i$ ,  $i = n - p + 1 : n$ . Como o espectro de  $A$  é simples e não contém o zero, conclui-se que  $Y^H AX^*$  é inversível e diagonalizável. Observe*

que o espectro de  $(I - XY^H)A$  é constituído de zero, com multiplicidade  $p$ , e dos outros  $n - p$  autovalores de  $A$ . Portanto, o espectro de  $F'(X^*)$ , que é igual a  $\{\lambda_i - \lambda_j | i = 1 : n - p, j = n - p + 1 : n\} \cup \{-\lambda_j | j = n - p + 1 : n\}$ , não contém o zero e, assim, é inversível. Concluímos, então, que existe uma vizinhança  $B$  de  $X^*$  tal que, para todo  $X \in B$ ,  $F'(X)$  é inversível, e  $Y^HAX$  é inversível e diagonalizável. Os métodos apresentados a seguir pressupõem que estamos trabalhando nessa vizinhança  $B$ .

Supondo  $X_0 \in \mathcal{X}_Y$ , o método de Newton aplicado a esse problema é o seguinte: para  $k \geq 0$ ,

$$X_k = X_{k-1} + (F'(X_{k-1}))^{-1} F(X_{k-1}).$$

Pode-se verificar facilmente que, se  $X_0 \in \mathcal{X}_Y$ , então  $X_k \in \mathcal{X}_Y$ , para todo  $k \geq 1$  [2]. Em [3], O. H. Hald introduz um método para resolver o problema escalar  $f(x) = 0$ , sem que seja preciso inverter a derivada de  $f$  (como no método de Newton):

**Proposição 2.** *Seja  $f : \mathbb{R} \rightarrow \mathbb{R}$  uma função duas vezes diferenciável em torno de uma raiz simples  $\alpha$  de  $f$ . Então existe uma vizinhança  $B$  de  $\alpha$  tal que, dado qualquer  $x_0 \in B$ , se  $y_0$  for suficientemente próximo de  $(f'(\alpha))^{-1}$ , então o seguinte esquema converge a  $(\alpha, (f'(\alpha))^{-1})$ , pelo menos, quadraticamente:*

$$x_k = x_{k-1} - y_{k-1}f(x_{k-1}) \tag{3}$$

$$y_k = 2y_{k-1} - y_{k-1}f'(x_k)y_{k-1} \tag{4}$$

A generalização desse fato para o caso de funções vetoriais também seria provado, mas com detalhes, em [5]:

**Proposição 3.** *Seja  $f : \mathbb{C}^n \rightarrow \mathbb{C}^n$  analítica. Seja  $\alpha \in \mathbb{C}^n$  tal que  $f(\alpha) = 0$  and  $f'(\alpha)$  é inversível. Então existe uma vizinhança  $B$  de  $\alpha$  tal que, dado qualquer  $x_0 \in B$ , se  $X_0$  é suficientemente próximo de  $(f'(\alpha))^{-1}$ , então o seguinte esquema converge a  $(\alpha, (f'(\alpha))^{-1})$ , pelo menos, quadraticamente:*

$$x_k = x_{k-1} - X_{k-1}f(x_{k-1}) \tag{5}$$

$$X_k = 2X_{k-1} - X_{k-1}f'(x_k)X_{k-1} \tag{6}$$

Observe que o esquema acima é uma iteração de ponto fixo aplicada a

$$G(x, X) = \begin{pmatrix} x - Xf(x) \\ 2X - Xf'(x)X \end{pmatrix},$$

e o resultado segue do fato que  $G'(\alpha, (f'(\alpha))^{-1}) = 0$  (ver [5]). É digno de nota o fato que não há resolução de sistemas no método, só operações algébricas.

Vamos agora dar uma formulação vetorial à função  $F(X) = AX - XY^HAX$ , usando a notação de produto de Kronecker [4], para podermos aplicar a proposição acima. Para isso, seja  $V_k = \text{vec}(X_k)$ , isto é,

$$(V_k)_{i+jn-n} = (X_k)_{i,j}, \quad i = 1, \dots, n, \quad j = 1, \dots, p.$$

Note que  $\text{vec}$  é um isomorfismo linear. Definimos, então:

$$f(V_k) = (I_p \otimes (A - X_kY^HA)) V_k,$$

isto é,

$$(f(V_k))_{i+jn-n} = (F(X_k))_{i,j}, \quad i = 1, \dots, n, \quad j = 1, \dots, p.$$

Note que  $vec^{-1}(f(V_k)) = F(X_k)$  e que  $f'(V_k)$  pode ser identificada a uma matriz quadrada:

$$f'(V_k) = \left[ I_p \otimes (A - X_k Y^H A) - (Y^H A X_k)^T \otimes I_n \right].$$

Observe que, para todo  $H \in \mathbb{C}^{n \times p}$ , se  $h = vec(H)$ , então

$$vec((F'(X_k))^{-1} H) = (f'(V_k))^{-1} h. \tag{7}$$

Agora, dados  $V_0, W_0 \in \mathbb{R}^{np}$ , o esquema de Moser-Hald se escreve como

$$V_{k+1} = V_k - W_k f(V_k) \tag{8}$$

$$W_{k+1} = 2W_k - W_k f'(V_{k+1}) W_k. \tag{9}$$

Gostaríamos que, sob algumas condições, os vetores  $V_k$  resultantes das iterações dadas pelo esquema de Moser-Hald permanecessem em  $x_Y = \{V \in \mathbb{R}^{np}; Y^H vec^{-1}(V) = I\}$ . Isso é o que será mostrado a seguir.

**Proposição 4.** *Se  $V_0 \in x_Y$  e  $W_0 = (f'(V_0))^{-1}$ , então  $V_k \in x_Y$  para todo  $k \geq 0$ .*

**Prova.** Seja  $Y^\perp = \{Z \in \mathbb{C}^{n \times p}; Y^H Z = 0\}$ . Para mostrar essa proposição, vamos mostrar por indução em  $k$  que  $V_k \in x_Y$  e, para todo  $Z \in Y^\perp$ ,  $vec^{-1}(W_k vec(Z)) \in Y^\perp$ . Assim, começamos com  $V_0 \in x_Y$  e com qualquer  $Z \in \mathbb{C}^{n \times p}$  tal que  $Y^H Z = 0$ . Seja  $V \in \mathbb{C}^{n \times p}$  tal que  $F'(X_0)V = Z$ . Então,

$$0 = Y^H F'(X_0)V = Y^H (AV - X_0 Y^H AV - V Y^H A X_0) = -Y^H V Y^H A X_0.$$

Como  $Y^H A X_0$  é inversível,  $Y^H V = 0$ , ou seja,  $Y^H vec^{-1}(W_0 vec(Z)) = 0$ .

Vamos supor, por indução, que  $V_k \in x_Y$  e que, para todo  $Z \in Y^\perp$ ,  $Y^H vec^{-1}(W_k vec(Z)) = 0$ . Então, como  $Y^H F(X_k) = 0$ , temos que  $Y^H vec^{-1}(W_k f(V_k)) = 0$ . Logo,  $X_{k+1} = vec^{-1}(V_{k+1}) \in x_Y$  e, assim,

$$Y^H F'(X_{k+1}) vec^{-1}(W_k vec(Z)) = 0,$$

pois

$$\begin{aligned} & Y^H F'(X_{k+1}) vec^{-1}(W_k vec(Z)) = \\ & = Y^H (A vec^{-1}(W_k vec(Z)) - X_{k+1} Y^H A vec^{-1}(W_k vec(Z)) - vec^{-1}(W_k vec(Z)) Y^H A X_{k+1}) = \\ & = Y^H A vec^{-1}(W_k vec(Z)) - (Y^H X_{k+1}) Y^H A vec^{-1}(W_k vec(Z)) - \\ & \quad - (Y^H vec^{-1}(W_k vec(Z))) Y^H A X_{k+1}. \end{aligned}$$

Ou seja,  $F'(X_{k+1}) vec^{-1}(W_k vec(Z)) \in Y^\perp$  e, portanto, por hipótese de indução,

$$Y^H vec^{-1}(W_k vec(F'(X_{k+1}) vec^{-1}(W_k vec(Z)))) = 0.$$

Mas,  $vec^{-1}(W_k vec(F'(X_{k+1}) vec^{-1}(W_k vec(Z)))) = vec^{-1}(W_k f'(V_{k+1}) W_k vec(Z))$ . Logo,

$$\begin{aligned} Y^H vec^{-1}(W_{k+1} vec(Z)) & = Y^H vec^{-1}((2W_k - W_k f'(V_{k+1}) W_k) vec(Z)) = \\ & = 2Y^H vec^{-1}(W_k vec(Z)) - Y^H vec^{-1}(W_k f'(V_{k+1}) W_k vec(Z)) = 0. \end{aligned}$$

□

## Resultados Numéricos

Uma classe de métodos que computam uma decomposição parcial do espectro de uma matriz  $A$  é a dos métodos de iteração simultânea. Esses métodos se iniciam com um conjunto l.i. de vetores que são transformados, a cada passo, em outros vetores via algum operador que, em geral, é uma função analítica de  $A$  cuja implementação seja viável, do ponto de vista computacional. Por exemplo, se é possível resolver sistemas lineares com uma matriz  $A$ , então operadores do tipo  $(aA - bI)^{-1}$  são viáveis. Os métodos de iteração simultânea incluem ainda algum procedimento que desvie da tendência à dependência linear os vetores resultantes, por exemplo, uma decomposição QR. Vamos criar aqui um novo método de iteração simultânea de potência (isto é, quando o operador é  $A$ ), partindo-se de uma matriz de vetores  $X_0$  tal que  $Y^H X_0 = I$ . O procedimento de desvio será o seguinte: se  $Z_k = AX_k$  é a matriz dos vetores resultantes da potência no passo  $k$ , então  $X_{k+1} = Z_k(Y^H Z_k)^{-1}$ . Assim, não só os vetores resultantes continuam na variedade como também permanecem l.i.. Os autovalores são então calculados a partir da decomposição espectral da matriz  $Y^H AX_N$ , em que  $N$  é o número de passos. Vamos comparar primeiro os dois métodos apresentados aqui, o de Newton e o de Newton-Moser, com esse método de iteração simultânea, partindo-se do mesma matriz inicial  $X_0$ , com  $Y^H X_0 = I$ . Nos testes,  $X_0 = Y = eye(n, p)$ , isto é, a matriz formada pelas  $p$  primeiras colunas da identidade de ordem  $n$ . Em todos os métodos, escolhemos  $N = p = 10$ . Tomamos emprestadas as matrizes *lesp* da galeria de matrizes do MATLAB:  $A = lesp(n)$ ,  $n = 50, 100, 200$ . Essas matrizes, embora tridiagonais e simétricas, possuem autovalores bastante sensíveis a perturbação das entradas, ou seja, o problema de autovalores dessas matrizes é mal condicionado (ver <http://www.mathworks.com/help/matlab/ref/gallery.html>). Computando autovalores em MATLAB via função *eig*, os 47 maiores autovalores de *lesp*(50), *lesp*(100) e *lesp*(200) coincidem, assim como os 97 maiores autovalores de *lesp*(100) e *lesp*(200). O interessante foi que cada um dos três métodos computou os mesmos dez maiores autovalores para  $n = 50, 100, 200$ , mesmo quando errava, como no caso do método de potências simultâneas. Esses autovalores estão expostos, por isso, numa única tabela: a Tabela 1. O método de Newton-Moser teve problemas de convergência, embora nas quatro primeiras iterações o método convergisse. Mas, da quinta iteração em diante, o método não convergia para o décimo maior autovalor, que ia para infinito. Ou seja, a matriz inicial  $W_0$  não estava na zona de convergência do método. Fizemos, então, o seguinte: reiniciamos parcialmente o método na quarta interação:  $W_4 = (f'(V_3))^{-1}$ . Com isso, o método de Newton-Moser convergiu otimamente, mas despendeu um tempo considerável. Os tempos médios (relativos a cinco performances sucessivas), em segundos, obtidos pelos três métodos estão na Tabela 2. Fizemos testes com os métodos de Newton e Newton-Moser, aplicados dessa vez às matrizes de Moler, também da galeria de matrizes testes do MATLAB:  $A = moler(n)$ ,  $n = 50, 100, 200$ . Essas matrizes são simétricas definidas positivas, mas têm um autovalor muito próximo de zero. O comando  $A = gallery('moler', n, \alpha)$  calcula  $U' * U$ , em que  $U = gallery('triu', n, \alpha)$ , a matriz triangular superior com 1 na diagonal e  $\alpha$  em todas as superdiagonais. Por *default*,  $\alpha = -1$ . Nesse caso,  $A(i, j) = \min(i, j) - 2$  e  $A(i, i) = i$ . O método de Newton-Moser não convergiu quando aplicado a essas matrizes. Notávamos que nove entre dez autovalores pareciam convergir, mas eram contaminados pelo décimo, que crescia ou decrescia infinitamente. Tentamos o mesmo procedimento descrito acima, mas o *restart* não parecia influenciar positivamente na convergência do método, mesmo ele sendo aplicado depois de números de passos diferentes. Mesmo com dois *restarts*, nada parecia fazer que o método convergisse para dez autovalores da matriz  $A$ . O décimo autovalor continuava divergindo de forma aberrante e não pudemos evitar essa contaminação nos testes.

Newton	Newton-Moser	Potências	QR (Eig)
-4.5491	-4.5491	-4.5491	-4.5491
-6.9531	-6.9531	-6.9531	-6.9531
-8.9979	-8.9979	-8.9979	-8.9979
-11.0000	-11.0000	-11.0000	-11.0000
-13.0000	-13.0000	-13.0000	-13.0000
-15.0000	-15.0000	-15.0001	-15.0000
-17.0000	-17.0000	-17.0018	-17.0000
-19.0000	-19.0000	-19.0251	-19.0000
-21.0000	-21.0000	-21.1979	-21.0000
-23.0000	-23.0000	-23.9318	-23.0000

Tabela 1: Os dez maiores autovalores de  $lesp(n)$ ,  $n = 50, 100, 200$

Método	$n = 50$	$n = 100$	$n = 200$
Newton	0.22	1.66	12.6
Newton-Moser	0.46	17.0	222
Potências	0.00093	0.0019	0.0083

Tabela 2: Tempos despendidos pelos três métodos

$n = 50$		$n = 100$		$n = 200$	
Newton	QR (Eig)	Newton	QR (Eig)	Newton	QR (Eig)
9.9983	9.9983	35.2420	34.6003	32.2805	32.2805
4.2738	4.2738	6.7584	6.7760	5.4462	5.4462
3.0944	3.0991	4.2325	4.2251	3.6574	3.6573
2.9577	2.9820	2.6500	2.6427	2.7759	2.7759
2.5267	2.5374	2.5155	2.5180	2.5171	2.5171
2.4162	2.4230	2.3784	2.3741	2.3913	2.3913
2.3152	2.3154	2.3153	2.3152	2.3176	2.3176
2.2766	2.2766	2.2760	2.2765	2.2779	2.2779
2.2563	2.2563	2.2562	2.2563	2.2570	2.2569
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Tabela 3: Dez autovalores de  $moler(n)$ ,  $n = 50, 100, 200$

## Conclusões

Os objetivos principais deste trabalho foram: mostrar como aproximar a inversa da derivada de uma função vetorial no método de Newton sem perder sua convergência quadrática; apresentar resultados práticos de um método de Newton que é praticamente inexplorado; introduzir uma versão aproximada desse método, inédito na literatura de computação de autovalores. Um arcabouço do nosso método já havia sido ligeiramente apresentado em um congresso no CEP-PEL/Eletróbrás em 5 de dezembro de 2011 (ver [1]). Porém, na época, não sabíamos ainda que

a dinâmica das suas iterações gerava vetores na variedade linear e, por manipulações algébricas, obrigávamos, então, os vetores a permanecerem na variedade. A proposição 4 é inédita e por ela então ficamos sabendo que, sob algumas condições iniciais, os vetores gerados pelo método se mantêm na variedade. Um problema com esse método, que tem surgido nos testes numéricos que realizamos com diversas matrizes, é que ele depende bastante das condições iniciais. Por outro lado, esse método pode ser importante no caso de nos depararmos com uma derivada (uma matriz) que resulte em um sistema difícil de se resolver. Se não pudermos iniciar o procedimento com  $W_0 = (f'(V_0))^{-1}$ , a necessidade de boas condições iniciais se torna ainda mais dramática. Na busca de melhores condições iniciais para o método de Newton-Moser aplicado às matrizes de *lesp* fizemos o seguinte: depois de 3 passos, reinicializamos o método com  $W_4 = (f'(V_3))^{-1}$ . Por outro lado, o método de Newton teve uma boa performance com essas matrizes, convergindo sem problemas, conforme a Tabela 1. As matrizes de *lesp* não possuem os autovalores muito próximos um do outro, o que faz com que o espectro de  $F'(X^*)$  não tenha nenhum autovalor próximo de zero. Agora, as matrizes de Moler têm autovalores muito próximos um do outro e, além disso, têm um autovalor muito próximo de zero: a função *eig* do MATLAB calculou  $-1.1484e-015$  (sic), para  $n = 50$ ;  $3.2752e-015$ , para  $n = 100$ ;  $9.3068e-016$ , para  $n = 200$ . Os testes com as matrizes de Moler confirmaram o que esperávamos de antemão e a convergência do Método de Newton, em geral, não foi boa, conforme podemos ver na Tabela 3. Porém, incrivelmente, o método funcionou melhor para  $n = 200$ . Aplicado às matrizes de Moler, o método de Newton-Moser não convergiu, embora, como já foi comentado anteriormente, nove dos dez valores encontrados fossem próximos de autovalores verdadeiros. Mas o décimo ia crescendo (ou decrescendo) infinitamente. Uma solução para isso talvez seja implementar um processo de deflação, que retira do processo os valores convergidos e respectivos autovetores, isto é, uma vez que algum autovalor convergir, a dimensão  $p$  do problema diminui de um. O problema é que, se não fizermos isso de modo eficiente, o método poderá convergir para valores já obtidos. Estamos com resultados promissores nesse sentido, mas isso é tema para trabalho futuro, pois a deflação demanda um tempo ainda maior e precisamos lidar melhor com isso. Em [5], há um resultado que garante a convergência do método, em aritmética exata, se  $\|f'(V^*)V_k - I\| < 1$  a partir de certo  $k$  ( $V^* = \text{vec}(X^*)$ ). Em [1], podemos ver outros resultados numéricos obtidos com o método de Newton, dessa vez aplicado a uma matriz de ordem 1664, uma matriz de estado proveniente do sistema elétrico de potência brasileiro. A convergência foi ótima, porque os autovalores são suficientemente afastados um do outro. Devemos observar que, em [1], seguimos a implementação sugerida em [2], sem produto de Kronecker. Aqui foi feita uma implementação inédita e contemporânea, usando produto de Kronecker, que é bem suportada pelas novas arquiteturas de computadores (no caso da *lesp(20)*, a versão via produto de kronecker resulta em matrizes de ordem 2000). Quanto ao método de potências simultâneas, é um método que demora a convergir, e isso já é bem conhecido na literatura. Porém, por outro lado, seu custo computacional é extremamente pequeno. Em geral, os métodos de iteração simultânea, assim como o método de Arnoldi, computam autovalores em regiões do plano complexo. A filosofia dos métodos apresentados aqui é diferente, pois buscamos autovetores em uma dada variedade. O que fizemos aqui, foi criar e testar um método de iteração simultânea adaptado a essa nova demanda. Por acaso, os autovetores das matrizes de *lesp* que estão na variedade, cujos vetores tem as  $p$  primeiras coordenadas livres (que foi a nossa variedade escolhida), são associados aos maiores autovalores negativos (menores valores em valor absoluto). Finalmente, os tempos de computação para esse trabalho foram obtidos via comandos *tic - toc* do MATLAB R2011b, rodando em um computador HP Compaq 7000 Pro AiO, com processador Intel Core(TM)2 Duo CPU E8400 3.00 GHz.

## Referências

- [1] L. H. Bezerra, Fixed-Point Methods in Eigenvalue Calculation, em “Workshop on Eigenvalue Computation and other Numerical Linear Algebra Problems: in Honor of Prof. B. Parlett”, [http://www2.cepel.br/workshop\\_eigen/DownloadPresentations/CepelWorkshopLicioBezerra.pdf](http://www2.cepel.br/workshop_eigen/DownloadPresentations/CepelWorkshopLicioBezerra.pdf), 2011.
- [2] F. Chatelin, Simultaneous Newton’s Iteration for the Eigenproblem, em “Defect Correction Methods, Theory and Applications (Computing Supplementum 5)” (K. Böhmer e H. J. Stetter, eds.) pp. 67-74, Springer-Verlag, Berlin, 1984.
- [3] O. H. Hald, On a Newton-Moser Type Method, *Numer. Math.*, 23 (1975) 411-426.
- [4] R. A. Horn e C. R. Johnson, “Topics in Matrix Analysis”, Cambridge University Press, Cambridge, 1991.
- [5] R. McGehee, A Note on the Moser-Hald Variation of Newton’s Method, em “Analysis, et cetera: Research Papers Published in Honor of Jurgen Moser’s 60th Birthday” (P. Rabinowitz e E. Zehnder, eds.), pp. 495-499, Academic Press, Boston, 1990.